

Programação Paralela e Distribuída

Prática 4: Python multiprocessing

(<https://docs.python.org/3/library/multiprocessing.html>)

1 – Considere o exemplo da T03 (pp. 31-34). Modifique o exemplo de forma a poder também executar e medir tempos para o caso usar processos.

Estudar os tempos de execução das 3 funções analisadas (ver tabela abaixo), lançadas:

i) Sequencialmente,

ii) Como Threads

iii) Como processos independentes.

a) Compare a execução Sequencial, execução com Threads e execução com Processos para 1, 2, 4 e 8 execuções de cada situação. Construa uma tabela com os tempos de execução e analise os resultados.

b) Repare que nos exemplos anteriores não está a dividir a tarefa a executar em várias sub-tarefas, está apenas a executar o código da função sequencial várias vezes, em sequência, com várias Treads a executar o mesmo e com vários processos a fazer o mesmo.

- Redefina o exercício, para as funções F2 e F3, de forma a poder comparar tempos de execução em termos de fazer o trabalho sequencialmente ou dividir esse mesmo trabalho em Threads ou em Processos. Comece por aumentar a dimensão dos problemas.

Funções exemplo:

F1	F2	F3
<pre>def function_to_run(): #Fibonacci a = 0 b = 1 for i in range (100000): a = b b = a + b</pre>	<pre>def function_to_run(): for i in range(10): with urllib.request.urlopen ("https://google.com")as f: f.read(1024)</pre>	<pre>def function_to_run(): file = open ("test.dat", "rb") size = 1024 for i in range (1000): file.read(size)</pre>

Para cada exercício que se segue, compare tempos de execução (sequencial versus paralelo e várias implementações em paralelas).

2 - Suponha que se pretende uma aplicação que leia linhas de texto de um ficheiro, as converta para maiúsculas e finalmente as escreva para um segundo ficheiro. A leitura do primeiro ficheiro deve ser feita por um processo (p1) a conversão da linha de texto e a escrita para o ficheiro resultado deve ser feita por outro processo (p2).

a) Usar uma Queue para comunicação entre processos.

b) Altere o programa para em vez de converter as linhas de texto para maiúsculas, o processo p2, encriptar a linha de texto (usando um algoritmo à sua escolha).

c) Implemente uma versão sequencial e compare os tempos de execução.

3 – Construa um programa que usa um processo para gerar matrizes de valores aleatórios e um segundo processo para calcular o quadrado dessas matrizes. Cada matriz ao quadrado deve ser devolvida ao primeiro processo que a deve escrever num ficheiro. Usar uma Queue para comunicação entre processos.

4 – Implemente uma versão sequencial, e compare tempos de execução, variando a dimensão das matrizes e o número de matrizes geradas.